

<b>Noname manuscript No.</b> (will be inserted by the editor)
--

---

# Object recognition in noisy RGB-D data using GNG

Received: date / Accepted: date

**Abstract** Object recognition in 3D scenes is a research field in which there is intense activity guided by the problems related to the use of 3D point clouds. Some of these problems are influenced by the presence of noise in the cloud that reduces the effectiveness of a recognition process. This work proposes a method for dealing with the noise present in point clouds by applying the Growing Neural Gas (GNG) network filtering algorithm. This method is able to represent the input data with the desired number of neurons while preserving the topology of the input space. The GNG obtained results which were compared with a Voxel Grid filter to determine the efficacy of our approach. Moreover, since a stage of the recognition process includes the detection of keypoints in a cloud, we evaluated different keypoint detectors to determine which one produces the best results in the selected pipeline. Experiments show how the GNG method yields better recognition results than other filtering algorithms when noise is present.

**Keywords** 3D Object Recognition · Growing Neural Gas · Keypoint Detection

## 1 Introduction

3D object recognition is a growing research field which has been stimulated by the well-known advantages offered by the use of 3D sensors compared against 2D based recognition methods. However, there are several difficulties to be

---

José Carlos Rangel · Vicente Morell · Miguel Cazorla · Sergio Orts-Escolano · José García-Rodríguez  
Institute for Computer Research. University of Alicante, Spain  
Tel.: +34-965903900  
Fax: +34-965903902  
E-mail: jcrangel@dccia.ua.es, vmorell@dccia.ua.es, miguel.cazorla@ua.es, sorts@dtic.ua.es, jgarcia@dtic.ua.es

overcome in order to achieve effective recognition. Some of these difficulties are: noise, occlusions, rotations, translations, scaling or holes that are present in the raw 3D point clouds provided by current RGB-D sensors such as Microsoft Kinect. Therefore, new algorithms are required to handle these problems when performing a correct object recognition process.

There exist several previous works in the field of 3D object recognition. Some of them provide a survey, review or evaluation of the existing 3D object recognition methods, while other works focus on the proposal of new methods and approaches for the recognition process. In [8], a survey of 3D object recognition methods based on local surface features is presented. They divide the recognition process into three basic stages: 3D keypoint detection, feature description, and surface matching. It also describes existing datasets and algorithms used in each stage of the whole process. Other studies, such as [23], focus on the evaluation of stereo algorithms. It presents an evaluation in terms of the recognition ability of this kind of algorithms. Using a different approach, [2] evaluates the different 3D shape descriptors for object recognition to study their feasibility in 3D object recognition.

There are some works that propose novel object recognition pipelines, such as [9], which combines depth maps and images, achieving good recognition results for heavily cluttered scenes. In [22], a novel Hough voting algorithm is proposed to detect free-form shapes in a 3D space, and this too produces good recognition rates. [17] describes a general purpose 3D object recognition framework that combines machine learning procedures with 3D local features, without a requirement for a priori object segmentation. This method detects 3D objects in several 3D point cloud scenes, including street and engineering scenes. [1] proposes a new method called Global Hypothesis Verification (Global HV), which is added to the final phase of the recognition process to discard false positives. Our approach is based on the pipeline presented in that work, introducing noise into the original point cloud to test the effect of that noise on the recognition process.

To make the object recognition system robust to noise, we propose the use of a Growing Neural Gas (GNG) [7] to represent and reduce the raw point clouds. This self-organizing map learns the distribution of the input space, adapting its topology. This feature makes it possible to obtain a compact and reduced representation of the input space in a set of 3D neurons and their connections. In addition, we test different keypoint detectors to determine which one obtains better recognition results. This GNG reduction improves the recognition process and reduces noisy 3D values. GNG has been used previously in [27] to filter and reduce point clouds. We also compare our proposal against other reduction/filtering methods, such as Voxel Grid. Hence, we present experiments that test a 3D object recognition pipeline with both the raw point cloud, the GNG and Voxel Grid filtered point clouds.

The rest of this work is organized as follows. First, in Section 2 we introduce and describe the GNG and Voxel Grid methods that we will use in the experimentation. Then, in Section 3 the pipeline is explained. Section 4 describes the dataset used in the experiments and how the recognition exper-

iments are carried out. After that, in Section 5 we present the results and a discussion of our experiments and, finally, in Section 6 conclusions and further work are presented.

## 2 3D Filtering methods

One way of selecting points of interest in 3D point clouds is to use a topographic mapping where a low dimensional map is fitted to the high dimensional manifold of the model, whilst preserving the topographic structure of the data. In this section, we review some typical methods for representing and reducing 3D data. First, we describe the Growing Neural Gas algorithm and how it works. Then, we briefly describe the Voxel Grid method, which is another commonly used data structure, in order to compare our proposed method.

### 2.1 GNG method

A common way to achieve a multi-dimensional reduction is by using self-organizing neural networks where input patterns are projected onto a network of neural units such that similar patterns are projected onto units adjacent in the network and vice versa. As a result of this mapping, a representation of the input patterns is achieved that, in the post-processing stages, makes it possible to exploit the similarity relations of the input patterns. However, most common approaches are not able to provide good neighborhood and topology preservation if the logical structure of the input pattern is not known a priori. In fact, the most common approaches specify in advance the number of neurons in the network and a graph that represents topological relationships between them, for example a two-dimensional grid, and seek the best match to the given input pattern manifold. When this is not the case, the networks fail to provide good topology preservation, as for example in the case of Kohonen's algorithm [12]. The approach presented in this paper is based on self-organizing networks trained using the Growing Neural Gas learning method [7], which is an incremental training algorithm. The links between the neurons in the network are established through competitive Hebbian learning [14]. As a result, the algorithm can be used in cases where the topological structure of the input pattern is not known a priori, and yields topology preserving maps of the feature manifold [15]. The main difference with respect to the original method [7] is that in our method a neuron is composed of 3 data elements ( $X, Y, Z$ ) representing the point coordinates.

In GNG, nodes in the network compete to determine the set of nodes with the highest similarity to the input distribution. In our case, the input distribution is a finite set of 3D points that can be extracted from different types of sensors. The highest similarity reflects which node, together with its topological neighbors, is the closest to the input sample point, which is the signal generated by the network. The  $n$ -dimensional input signals are randomly generated from a finite input distribution.

The nodes move towards the input distribution by adapting their position to the input data geometry. During the learning process local error measures are gathered to determine where to insert new nodes. New nodes are inserted near the node with the highest accumulated error. At each adaptation step a connection between the winner and its topological neighbors is created as dictated by the competitive Hebbian learning rule. This is continued until an ending condition is fulfilled, as for example evaluation of the optimal network topology, a predefined network size or a deadline.

The network is specified as:

- A set  $N$  of nodes (neurons). Each neuron  $c \in N$  has its associated reference vector  $w_c \in R^d$ . The reference vectors can be considered as positions in the input space of their corresponding neurons.
- A set of edges (connections) between pairs of neurons. These connections are not weighted and their purpose is to define the topological structure. An edge aging scheme is used to remove connections that are invalid due to the motion of the neuron during the adaptation process.

The GNG learning algorithm is presented in Algorithm 1, and Figure 1 shows the working scheme for the GNG method. Using a Growing Neural Gas model to represent 3D data has some advantages over traditional methods such as Voxel Grid. For example, we specify the number of neurons (representative points of the map), while other methods such as Voxel Grid obtain a different number of occupied cells depending on the distribution and resolution of the cells (voxels). In our experiments, we have used the following parameters for the GNG method:  $\lambda = 2000$ ,  $\epsilon_w = 0.1$ ,  $\epsilon_n = 0.001$ ,  $\alpha = 0.5$  and  $\alpha_{max} = 250$ . These produced good results in our previous works with GNG and 3D data.

Figure 2 shows an example of a GNG representation of one of the objects we use in the experimentation. The GNG forms a map and we use only the neurons as the new, filtered and reduced representation of the object.

## 2.2 Voxel Grid method

The Voxel Grid (VG) down-sampling technique is based on the input space sampling using a grid of 3D voxels [28]. The VG algorithm defines a voxel grid in the 3D space and for each voxel, a point is chosen as the representative of all the points that lie on that voxel. It is necessary to define the size of the voxels as this size establishes the resolution of the filtered point cloud, and therefore the number of points that form the new point cloud. The representative of each cell is usually the centroid of the voxel’s inner points or the center of the voxel grid volume. Thus, a subset of the input space is obtained that roughly represents the underlying surface.

The VG method presents the same problems as other sub-sampling techniques: it is not possible to define the final number of points which represent the surface; geometric information loss due to the reduction of the points inside a voxel; and sensitivity to noisy input spaces.

```

input : N-dimensional input data
output: N-dimensional map
1 Start with two neurons  $a$  and  $b$  at random positions  $w_a$  and  $w_b$  in  $R^d$ .
2 repeat
3   for  $patterns=0$  to  $\lambda$  do
4     Generate at random an input pattern  $\xi$  according to the data distribution
        $P(\xi)$  of each input pattern.
5     Find the nearest neuron (winner neuron)  $s_1$  and the second nearest  $s_2$ .
6     Increase the age of all the edges emanating from  $s_1$ .
7     Add the squared distance between the input signal and the winner neuron
       to a counter error of  $s_1$  such that:
           
$$\Delta error(s_1) = \|w_{s_1} - \xi\|^2 \quad (1)$$

8     Move the winner neuron  $s_1$  and its topological neighbors (neurons
       connected to  $s_1$ ) towards  $\xi$  by a learning step  $\epsilon_w$  and  $\epsilon_n$ , respectively, of the
       total distance:
           
$$\Delta w_{s_1} = \epsilon_w(\xi - w_{s_1}) \quad (2)$$

9     forall the direct neighbors  $n$  of  $s_1$  do
10       |
           
$$\Delta w_{s_n} = \epsilon_n(\xi - w_{s_n}) \quad (3)$$

11     end
12     if  $s_1$  and  $s_2$  are connected by an edge then
13       | Set the age of this edge to 0.
14     else
15       | Create the connection between  $s_1$  and  $s_2$ .
16     end
17     Remove the edges larger than  $a_{max}$ 
18     if any neuron is isolated (without emanating edges) then
19       | Remove those neurons as well.
20     end
21 end
22 Insert a new neuron as follows:
23   Determine the neuron  $q$  with the maximum accumulated error.
24   Insert a new neuron  $r$  between  $q$  and its furthest neighbor  $f$ :
           
$$w_r = 0.5(w_q + w_f) \quad (4)$$

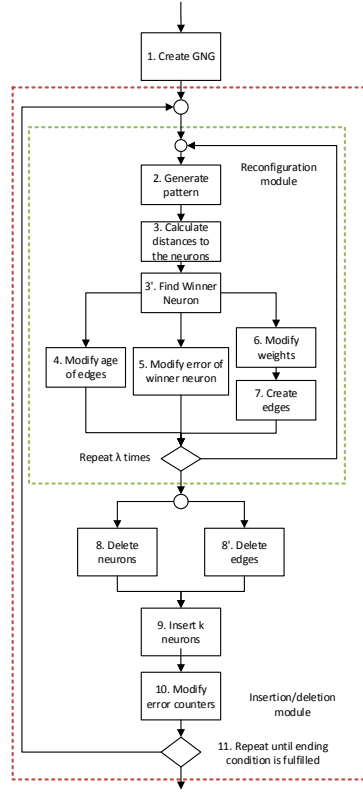
25   Insert new edges connecting the neuron  $r$  with neurons  $q$  and  $f$ , removing
       the old edge between  $q$  and  $f$ .
26   Decrease the error variables of neurons  $q$  and  $f$  multiplying them by a constant
        $\alpha$ . Initialize the error variable of  $r$  with the new value of the error variable of  $q$ 
       and  $f$ .
27   Decrease all error variables by multiplying them by a constant  $\gamma$ .
28 until number of neurons reached;

```

**Algorithm 1:** Pseudo-code of the GNG algorithm.

### 3 3D object recognition

This section describes the overall recognition pipeline, which is based on the one proposed in [1] (see Figure 4). Our proposal is based on local features from the point clouds, so our pipeline does not need a segmentation pre-stage. The recognition pipeline is explained in the following subsections.



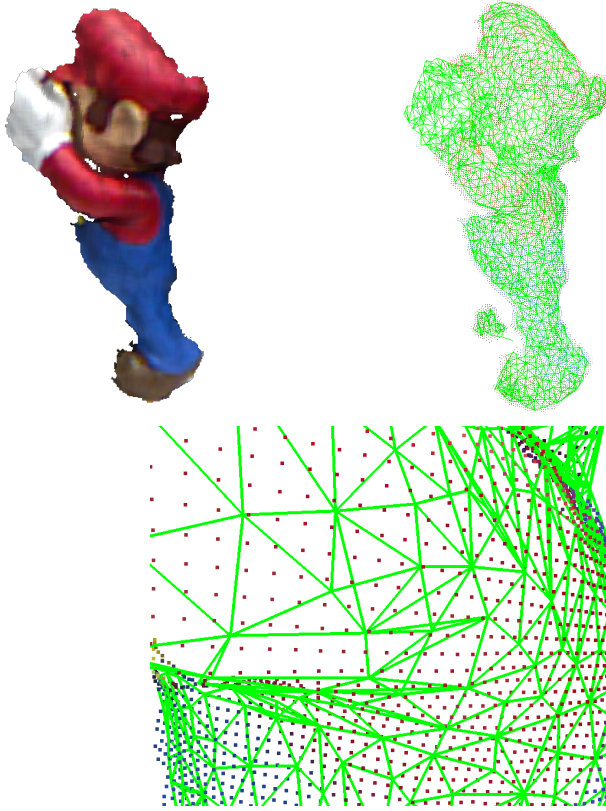
**Fig. 1** GNG scheme.

### 3.1 Normal extraction

Point normals are frequently used in many areas like computer graphics. Given a geometric surface, it is usually trivial to infer the direction of the normal at a certain point on the surface as the vector perpendicular to the surface at that point. Normals are used to infer the orientation of the surface in a coordinate system. Our system will use the normals computed both in the scene and the model to compute other necessary data such as descriptors and reference frames in the following steps [18].

### 3.2 Keypoint detection

After the normals are estimated, the next step is to extract the keypoints for the model and scene point cloud. With this stage we can reduce the number of points in the point cloud. It allows us to select only points that are representative of the cloud. This stage makes it possible to reduce the time required to describe the features in further stages. The selection of the keypoints is



**Fig. 2** Top left: Original object. Top right: GNG representation of one of the objects we use in the experimentation. Bottom: Zoom of the arm of Mario.

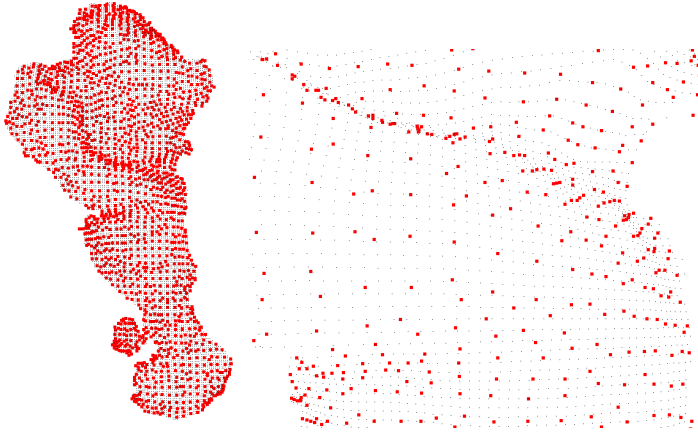
made using the following keypoint detectors: Uniform Sampling, Harris3D or the Intrinsic Shape Signature(ISS) method.

### 3.2.1 Uniform Sampling

This method builds a 3D grid over the input point cloud. This grid of 3D cuboids, which are called voxels, are located upon the point cloud and only one point is used to represent all the points inside each voxel [19]. This representative point is usually the centroid of the inner points inside a voxel.

### 3.2.2 Harris 3D

Harris 3D[8][21] is a robust point of interest detector for 3D meshes. It adapts the well-known 2D Harris corner detection for images in order to be used for 3D meshes. It has proven to be effective, obtaining high repeatability values. It uses a Gaussian function to smooth the derivative surfaces and mitigate the



**Fig. 3** Left: Voxel Grid representation of one of the objects we use in the experimentation. Right: Zoom of the arm of Mario.

effect of local deformations introduced by noise, holes, etc. It also proposes an adaptive neighborhood selection which improves feature detection.

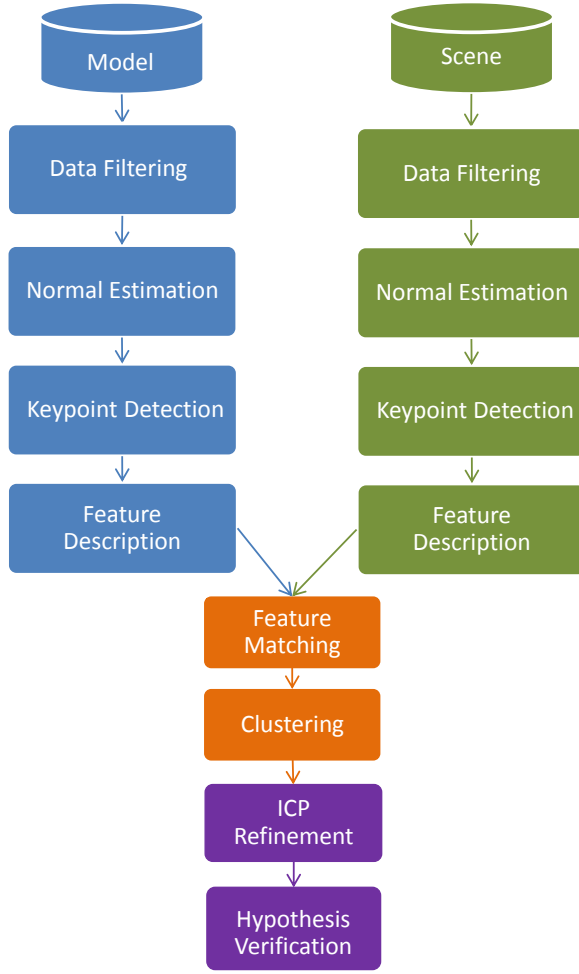
### 3.2.3 Intrinsic Shape Signatures(ISS)

The ISS[29][8][26] approach enables both highly discriminative shape matching and efficient pose estimation and registration for 3D point clouds. It is a detector that is carefully crafted to ensure discriminative, descriptive and robust keypoints in noisy scenes. It is based on the Eigenvalue Decomposition of the scatter matrix of the points belonging to a given neighborhood of a point. This method employs the ratio of two successive eigenvalues ( $\lambda_1, \lambda_2, \lambda_3$ ) to prune the points. Only the points whose ratio between two successive eigenvalues remains below a threshold ( $\tau$ ) are retained. Among the remaining points, the salience is determined by the magnitude of the smallest eigenvalue  $\lambda_1$ ,  $\lambda_2/\lambda_1 < \tau_{21}$  and  $\lambda_3/\lambda_2 < \tau_{32}$ .

## 3.3 Feature description

A descriptor codifies the underlying information in a certain neighborhood around a keypoint. Once the keypoints are computed, we need to extract the descriptors. The original work uses the Unique Signatures of Histograms for Local Surface Description (SHOT)[24] [8][6]. Here we test the pipeline using two other descriptors: Fast Point Feature Histograms (FPFH) [20] and Spin Image [11,10]. We briefly describe them below.





**Fig. 4** Recognition pipeline scheme.

### 3.3.1 Signatures of Histograms for Local Surface Description (SHOT)

This descriptor is an intersection between signatures and histograms. It takes each detected keypoint, builds a local reference frame and then divides the neighborhood space into 3D spherical volumes. Next, according to a function of the angle between the normal at the keypoint and the near points, a histogram is generated for each spherical volume by accumulating point counts in bins. Joining all the histograms makes the Signature of Histograms of Orientations (SHOT) descriptor. The SHOT descriptor is highly descriptive, computationally efficient and robust to noise.

### 3.3.2 Spin Image

The Spin Image feature descriptor [11,10] is calculated by placing a cylinder at every query point of the cloud, and orienting the cylinder with the normal at that point. Each cylinder is divided radially and vertically to create a set of volumes. Then, the descriptor is constructed by adding up the neighboring points that are inside each volume.

### 3.3.3 Fast Point Feature Histograms (FPFH)

FPFH [20] is a simplification of the PFH descriptor. To describe a point, the FPFH method computes a set of tuples between the query point and its neighbors. After using the same approach, the tuples are now computed for every selected neighbor and then these values are used to weight a final 16-bin histogram.

## 3.4 Feature matching

To determine the correspondences between model and scene descriptors, we used the KDTreeFLANN [16] method. This method of the FLANN (Fast Library for Approximate Nearest Neighbors) library uses a kd-tree and an approximate nearest neighbor scheme to find a close feature (the closest is not guaranteed) in a quick way. The structure searches for a scene feature in the model feature set, and only if the distance between the points is less than a threshold, the coordinates are considered as a correspondence and used in the next stage. This structure is commonly used due to its computational time improvement and its fast response results [13].

## 3.5 Clustering the features with the Geometric Consistency method

In this step, we group subsets of correspondences found in the above stage into smaller clusters by checking the geometric consistency of pairs of correspondences, using the Geometric Consistency (GC) grouping method. The GC algorithm assumes that the correspondences without geometric consistency will generate large errors in the transformations. Hence, with the geometric consistency method, it is possible to decrease the mismatched correspondences and improve the robustness of the hypothesized transformations [8] [4] [1].

## 3.6 ICP Refinement

This step refines the 6 DoF (Degrees of Freedom) pose by using the Iterative Closest Point (ICP) method. ICP is an iterative method which is able to find the transformation between two point clouds by using a dense representation of the point cloud [3] [5]. ICP uses as initialization the result provided by the clustering stage.

### 3.7 Hypothesis Verification

The Hypothesis Verification algorithm was proposed in [1]. This stage determines whether a given subset is a true or false positive. This method takes as input the instances found in the clustering stage and refined by the ICP method. Then, it uses a Simulated Annealing Meta-heuristic algorithm to solve the cost function used to determine whether the hypothesis is valid or not. The Hypothesis Verification method provides a set of instances of the model that match with the scene.

## 4 Experimentation

This section briefly describes the dataset used in the experimentation and how the experiments were performed.

### 4.1 Dataset

To test the different approaches, we used the SHOT Dataset<sup>1</sup> (University of Bologna)[25][24][6]. SHOT has been acquired by means of the Spacetime Stereo (STS) technique and consists of 7 models, with different views of each model, and 17 scenes for a total of 49 object recognition instances. Using these point clouds we will test the noise influence on the recognition rate. Figure 5 shows two models of the dataset and two scenes where they appear.

### 4.2 Experimentation setup

The experiment consists in searching a selected model in a scene with the recognition pipeline described above. As we are testing the method with different noise levels, we have applied five different levels of Gaussian noise with 0.001, 0.0025, 0.005, 0.0075 and 0.01 meters of standard deviation. We only applied noisy values in the scene point clouds as the stored models are supposed to have a higher quality level. These new clouds with noise, named the RAWNoise dataset, are reduced or filtered using the GNG and VoxelGrid methods. We apply the GNG algorithm to reduce the scene clouds to 10000, 15000, 17500 and 20000 representative points, which we call the GNGNoise dataset. This process is repeated using the Voxel Grid method to achieve a similar reduction in obtaining the VoxelGridNoise dataset. We repeat this process with each keypoint detector and descriptor.

In the end, we have a dataset with six different sets of point clouds and five levels of noise for the scene datasets. We decided to test every combination of model and scene on the different datasets to obtain the most representative values. Table 1 shows all the possible combinations of our datasets. The first

<sup>1</sup> <http://www.vision.deis.unibo.it/research/80-shot>



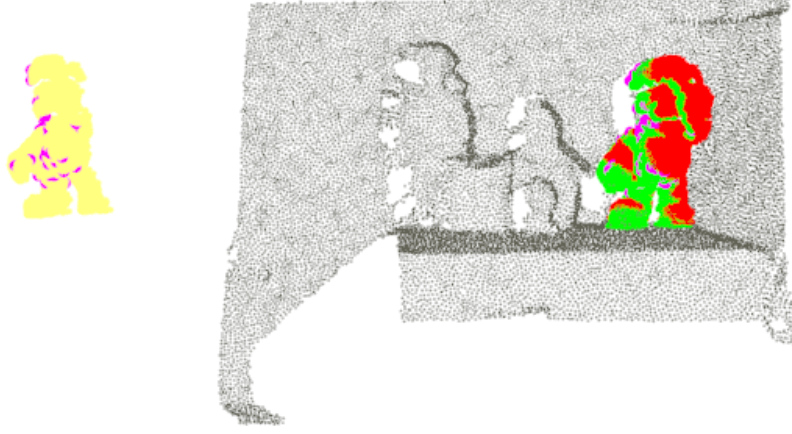
**Fig. 5** At top, two models of the dataset, namely those of Mario and Peter Rabbit. At bottom, two scenes from the dataset .

word of each pair is the method applied to the model and the second one is the scene method. We tested the system with the 49 recognition instances available in the dataset.

**Table 1** List of the experiment combinations.

<b>GNG Models</b>	<b>RAW Models</b>	<b>Voxel Grid Models</b>
GNG_GNG	RAW_GNG	VoxelGrid_GNG
GNG_GNGNoise	RAW_GNGNoise	VoxelGrid_GNGNoise
GNG_RAW	RAW_RAW	VoxelGrid_RAW
GNG_RAWNoise	RAW_RAWNoise	VoxelGrid_RAWNoise
GNG_VoxelGrid	RAW_VoxelGrid	VoxelGrid_VoxelGrid
GNG_VoxelGridNoise	RAW_VoxelGridNoise	VoxelGrid_VoxelGridNoise

To measure the performance of the recognition pipeline, we use the Hypothesis Verification algorithm, which analyzes the results and provides us with the true positives of the recognition method over the different datasets. When the system finds a true positive, it only takes the instance with the most matched points between the model and the scene that has been located, and shows a screen with the model superimposed on the scene, in the position



**Fig. 6** Recognition result obtained by the pipeline.

where the instance has been located (see Figure 6). The different colors for the model in the image indicate the following:

1. Green: a true positive instance recognized by the Hypothesis Verification algorithm.
2. Red: an instance aligned in the ICP Refinement step.
3. Yellow: Original model searched in the scene.
4. Purple: detected keypoints of the model.

## 5 Results

This section presents the results obtained after the execution of the different sets of experiments. We present the results for a descriptor and, then, for each detector.

### 5.1 Results for SHOT feature descriptor

Table 2 shows the percentage of true positives obtained using the Uniform Sampling keypoint detector with the SHOT descriptor. The number in the green shade represent the higher percentages, while yellow indicates the middle values and red the lowest ones. The recognition percentage obtained for a non-filtered scene and model was 86%, where we used all the points in the clouds,

obtaining the highest recognition value for all the experiments using Uniform Sampling. With this keypoint detector the better recognition results when adding noise were those filtered with the GNG method. In the presence of noise, the highest recognition rate obtained was 83%, using the scenes with the 0.005 noise level and filtered using the GNG method with 10000 representative points. On the other hand, the results for a raw cloud with the same noise level do not provide true positive recognition results. Comparing with the value obtained using scenes filtered with VG, GNG maintains the recognition rates. It is noteworthy that with a 0.01 noise level we did not obtain true positive recognition results in any experiment.

**Table 2** Results for the experiments using SHOT and Uniform Sampling.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	86	50	47	53	65	9	31	23	31
	0.001	82	45	49	50	57	17	23	27	28
	0.0025	20	58	67	74	73	9	16	22	36
	0.005	0	75	77	69	73	0	0	0	0
	0.0075	0	37	43	31	22	0	0	0	0
	0.01	0	3	0	0	0	0	0	0	0
GNG	0	84	52	58	57	70	23	29	26	42
	0.001	69	59	64	67	72	18	38	40	49
	0.0025	9	59	67	79	73	32	24	22	40
	0.005	0	83	71	73	76	0	0	3	0
	0.0075	0	49	31	14	10	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	78	51	66	64	70	27	38	42	47
	0.001	65	48	64	62	70	27	38	37	45
	0.0025	0	63	74	76	66	24	35	35	41
	0.005	0	60	74	63	69	0	0	3	8
	0.0075	0	36	24	26	13	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0

Table 3 shows the results for the Harris 3D keypoint detector. Using this detector the highest recognition rate was 81%, with a 0.005 noise level and filtered using the GNG method with 15000 representative points. For non-filtered scenes and models, the recognition percentage for this detector was 84%. The noisy scenes filtered by GNG obtain better results outperforming those filtered using the Voxel Grid method. Again, with this detector no true positive results were obtained for the cloud with a 0.01 noise level.

In Table 4 we show the results for the ISS keypoint detector. Using this detector the highest recognition value obtained was 74% , using a 0.001 noise level and filtered with GNG with 20000 representatives. Non-filtered scenes and models obtained 86% recognition, and for the scenes with a 0.01 noise level we did not obtain any true positive result.

Table 5 shows the mean of the recognition results obtained for the keypoint detectors evaluated. The highest values were obtained when using the scenes filtered by the GNG method. For the noisy clouds, the highest value obtained was 72% for the cloud filtered with the GNG method with 20000 representatives and a 0.001 noise level. GNG always obtains better results than Voxel Grid, and the number of representatives is an influential factor for recognition.

**Table 3** Results for the experiments using SHOT and Harris 3D.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	84	52	58	36	43	3	22	12	17
	0.001	80	49	51	48	51	3	14	15	23
	0.0025	0	63	57	54	74	9	21	31	44
	0.005	0	68	77	77	76	0	0	3	6
	0.0075	0	28	33	19	13	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
GNG	0	76	77	71	55	76	15	39	26	41
	0.001	58	64	66	67	68	19	32	40	50
	0.0025	0	69	77	70	77	21	34	53	49
	0.005	0	73	81	78	71	0	0	5	0
	0.0075	0	43	24	15	8	0	4	4	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	59	69	65	52	67	21	40	40	54
	0.001	42	64	64	66	72	19	52	47	50
	0.0025	0	64	63	67	70	15	43	65	64
	0.005	0	69	77	67	61	0	0	0	3
	0.0075	0	22	18	5	3	0	0	0	0
	0.01	0	0	0	0	6	0	0	0	0

**Table 4** Results for the experiments using SHOT and Intrinsic Shape Signature.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	86	32	39	55	50	0	9	13	33
	0.001	88	44	55	41	44	0	22	25	28
	0.0025	13	36	46	59	64	0	11	24	33
	0.005	0	29	56	44	50	0	0	0	0
	0.0075	0	8	3	0	0	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
GNG	0	80	28	51	56	56	7	20	24	31
	0.001	61	36	49	55	62	5	30	30	42
	0.0025	0	40	48	56	61	0	14	26	36
	0.005	0	24	48	39	34	0	0	0	0
	0.0075	0	6	0	0	0	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	80	26	47	50	63	22	47	39	43
	0.001	69	34	44	65	74	19	43	49	47
	0.0025	0	30	44	66	67	17	33	38	49
	0.005	0	22	39	39	33	0	0	0	0
	0.0075	0	0	4	0	0	0	0	0	0
	0.01	0	0	0	0	0	0	0	0	0

With the GNG filtered scenes we are able to recognize objects with a higher noise level.

Another noteworthy result is that GNG achieved a better rate than that obtained for the point clouds filtered with the Voxel Grid method. Table 6 shows the mean of all the recognition results, grouped by the numbers of representative points. The left part in this table represents the values achieved by the pipeline using the GNG filtered clouds and the right part shows the values obtained using the Voxel Grid method. In this table, we see that using the point clouds filtered by the GNG with 20000 representatives obtains better results than the one filtered one with fewer representatives.

Figure 7 shows the effect of the noise on the recognition process. The dotted line represents the experiments using the RawNoise scenes with the models indicated in the chart. Using a raw point cloud (noise level 0) scene,

**Table 5** Mean values of the results for the experiments using SHOT.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	85	45	48	48	53	4	21	16	27
	0.001	83	46	52	46	51	7	20	22	26
	0.0025	11	52	57	63	70	6	16	26	38
	0.005	0	57	70	63	66	0	0	1	2
	0.0075	0	24	26	17	12	0	0	0	0
	0.01	0	1	0	0	0	0	0	0	0
GNG	0	80	52	60	56	67	15	29	25	38
	0.001	63	53	60	63	67	14	33	37	47
	0.0025	3	56	64	68	70	18	24	34	42
	0.005	0	60	67	63	60	0	0	3	0
	0.0075	0	32	19	10	6	0	1	1	0
	0.01	0	0	0	0	0	0	0	0	0
VoxelGrid	0	72	49	59	56	67	23	42	40	48
	0.001	59	49	57	64	72	22	45	44	47
	0.0025	0	53	60	70	68	19	37	46	52
	0.005	0	50	63	56	55	0	0	1	4
	0.0075	0	19	16	10	5	0	0	0	0
	0.01	0	0	0	0	2	0	0	0	0

**Table 6** Mean values of the results for the experiments using the evaluated keypoint detectors grouped by the number of representatives.

	Scenes								
	RAW	GNG				VoxelGrid			
	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	30	38	42	39	42	3	9	11	16
GNG	24	42	45	43	45	8	15	17	21
VoxelGrid	22	37	43	43	45	11	21	22	25

recognition reaches the highest values but, adding different noise levels reduces this effectiveness even with low noise levels. These charts show how GNG outperforms the Voxel Grid method and helps to maintain the recognition process when noisy clouds are used. These results indicate that the algorithm only tolerates noise levels up to 0.0075 meters of standard deviation.

Calculating the mean of the recognition rates grouped by filtering method, Uniform Sampling obtains the highest true positive recognition rate with 47,8%, followed by Harris 3D with 46,8% and, finally, ISS with 31,2%.

## 5.2 Results for the FPFH feature descriptor

Tables 7, 8 and 9 show the results for Uniform Sampling, Harris3D and ISS, respectively. Table 10 shows the mean values grouped by the number of representatives. Finally, Figure 8 shows a comparison of the mean rate recognition results.

With this descriptor, although using the raw data the recognition rate is 92%, while in clouds with low noise the recognition rate decreases quickly. However, GNG is able to provide good recognition rates for noise levels up to 0.0025. For GNG, the recognition rate is better than the one provided by the SHOT descriptor. This behavior is the same for the three keypoint detectors. Regarding the mean values, GNG yields better results, although for different numbers of neurons.





**Fig. 7** Charts representing the mean results for different experiment sets.

Comparing the results obtained for each detector we can see that Harris 3D achieves the highest true positive mean recognition rate results for GNG, with 40,2%, Uniform Sampling obtained 39,8%, and ISS 28,8%.

**Table 7** Results for the experiments using FPFH and Uniform Sampling.

Models	Scene's Noise	Scenes									
		RAW All Points	GNG				VoxelGrid				
RAW	0	92	45	73	67	76	22	57	61	63	
	0.001	24	55	61	71	73	33	41	47	51	
	0.0025	2	43	65	84	80	33	35	39	37	
	0.005	2	47	51	37	18	4	4	0	2	
	0.0075	2	12	2	6	8	4	8	2	2	
	0.01	0	0	2	0	2	4	0	0	0	
GNG	0	80	47	80	73	80	39	53	61	63	
	0.001	14	53	61	63	76	51	59	69	59	
	0.0025	0	65	63	69	65	39	37	31	29	
	0.005	0	43	29	24	18	0	6	2	2	
	0.0075	0	2	0	2	0	2	0	2	2	
	0.01	8	0	0	0	0	4	2	4	0	
VoxelGrid	0	80	53	78	71	82	31	67	57	65	
	0.001	18	41	69	78	76	47	59	61	71	
	0.0025	0	47	65	67	82	27	39	39	47	
	0.005	0	45	47	29	22	8	0	0	0	
	0.0075	0	14	0	2	4	0	0	0	0	
	0.01	6	2	0	0	0	2	0	0	2	

**Table 8** Results for the experiments using FPFH and Harris 3D.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	90	57	65	65	65	33	35	37	45
	0.001	27	61	53	65	65	27	39	53	53
	0.0025	0	63	71	69	78	35	31	31	22
	0.005	0	61	51	43	18	8	4	6	2
	0.0075	2	10	10	4	4	2	2	2	2
	0.01	6	2	0	0	0	4	0	4	2
GNG	0	76	61	61	76	69	37	35	45	53
	0.001	14	55	67	67	65	43	45	43	43
	0.0025	2	67	73	67	69	31	24	31	14
	0.005	2	55	41	20	16	2	2	2	6
	0.0075	2	2	8	0	2	4	2	0	0
	0.01	6	2	2	0	4	0	0	6	0
VoxelGrid	0	73	55	78	63	63	31	63	47	55
	0.001	16	51	57	69	61	33	51	47	51
	0.0025	0	61	61	63	63	24	33	39	37
	0.005	6	59	45	45	20	2	6	4	8
	0.0075	2	12	10	8	4	4	0	2	2
	0.01	2	4	0	2	2	4	4	2	0

**Table 9** Results for the experiments using FPFH and Intrinsic Shape Signature.

		Scenes									
		RAW	GNG				VoxelGrid				
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	92	16	31	51	57	16	39	35	53	
	0.001	61	18	43	55	61	22	39	51	55	
	0.0025	20	27	65	61	65	20	31	37	31	
	0.005	8	29	29	18	20	0	2	2	0	
	0.0075	2	6	0	2	2	0	0	0	0	
	0.01	8	0	0	0	2	0	0	0	0	
GNG	0	84	33	57	57	63	24	55	43	51	
	0.001	49	35	61	55	73	22	45	51	57	
	0.0025	10	41	55	63	51	22	16	24	22	
	0.005	4	31	31	18	4	2	0	2	0	
	0.0075	0	2	0	4	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	
VoxelGrid	0	92	33	49	51	67	24	57	41	61	
	0.001	57	27	49	57	63	33	47	53	63	
	0.0025	12	29	53	55	69	16	33	35	41	
	0.005	4	18	20	22	16	0	0	0	0	
	0.0075	0	2	0	0	0	0	0	0	0	
	0.01	0	0	0	0	0	0	0	0	0	

### 5.3 Results for the Spin Image feature descriptor

Tables 12, 13 and 14 show the results for the experiments with the different keypoints using the Spin Image feature descriptor. We also have the same tables as the previous descriptors for the mean values (Table 15) and the mean values grouped by the number of representatives (Table 16).

For this set of experiments, GNG achieves better results than the Voxel Grid filtering method, but the recognition rates were very low. The best recognition rate was obtained by the ISS detector and the Spin Image descriptor. This indicates that the use of a filter method with the Spin Image descriptor is not appropriate. This is due to the fact that the Spin Image uses a number of the neighbors in a given region around the keypoint to build the descriptor. With the point reduction performed by GNG or Voxel Grid, the descriptor obtained could not be used for recognition tasks. Furthermore, this combi-

**Table 10** Mean values of the results for the experiments using FPFH.

Models	Scene's Noise	Scenes											
		RAW All Points	GNG				VoxelGrid						
			10000	15000	17500	20000	10000	15000	17500	20000			
RAW	0	91	39	56	61	66	24	44	44	54			
	0.001	37	45	52	64	67	27	39	50	53			
	0.0025	7	44	67	71	74	29	32	35	30			
	0.005	3	46	44	33	19	4	3	3	1			
	0.0075	2	10	4	4	5	2	3	1	1			
	0.01	5	1	1	0	1	3	0	1	1			
GNG	0	80	47	66	69	71	33	48	50	56			
	0.001	26	48	63	62	71	39	50	54	53			
	0.0025	4	58	64	67	62	31	26	29	22			
	0.005	2	43	33	21	13	1	3	2	3			
	0.0075	1	2	3	2	1	2	1	1	1			
	0.01	5	1	1	0	1	1	1	3	0			
VoxelGrid	0	82	47	68	62	71	29	63	48	61			
	0.001	31	39	59	68	67	37	52	54	62			
	0.0025	4	46	60	62	71	22	35	37	41			
	0.005	3	41	37	32	20	3	2	1	3			
	0.0075	1	10	3	3	3	1	0	1	1			
	0.01	3	2	0	1	1	2	1	1	1			

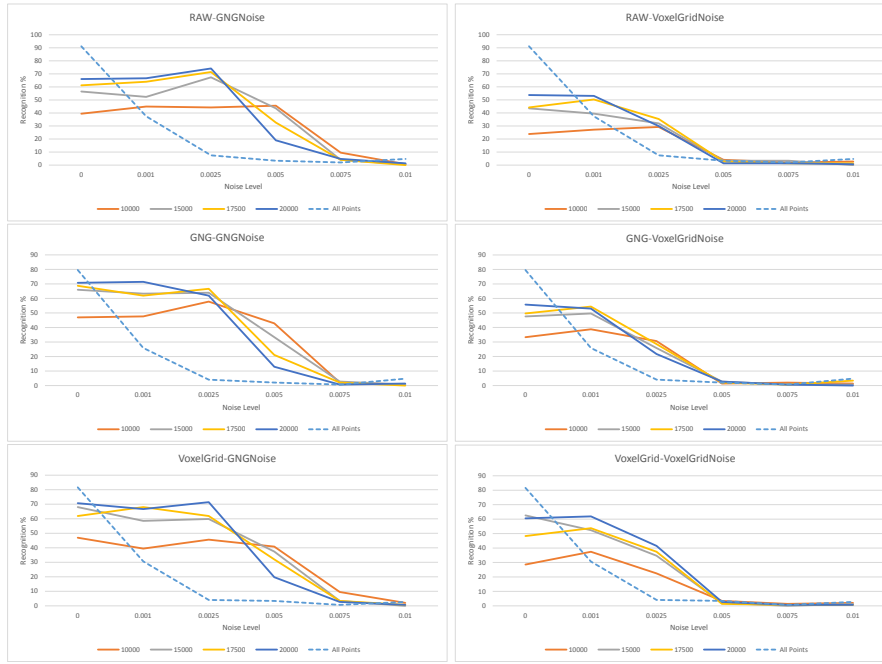
**Table 11** Mean values of the results for the experiments using the evaluated keypoint detectors with the FPFH descriptor, grouped by the number of representatives.

	Scenes								
	RAW	GNG				VoxelGrid			
	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	24	31	37	39	39	15	20	23	23
GNG	20	33	38	37	37	18	21	23	22
VoxelGrid	21	31	38	38	39	16	26	24	28

nation of detector and descriptor is computationally expensive, comparing it with Uniform Sampling and SHOT.

**Table 12** Results for the experiments using Spin Image and Uniform Sampling.

Models	Scene's Noise	Scenes											
		RAW All Points	GNG				VoxelGrid						
			10000	15000	17500	20000	10000	15000	17500	20000			
RAW	0	24	10	14	6	6	8	2	2	6			
	0.001	16	8	8	10	8	0	0	12	4			
	0.0025	6	2	14	16	10	2	0	4	4			
	0.005	0	8	2	6	12	4	4	4	10			
	0.0075	14	8	2	10	14	8	8	4	0			
	0.01	41	4	2	10	14	14	10	8	12			
GNG	0	18	6	12	8	14	6	2	0	4			
	0.001	10	6	12	12	10	0	2	8	2			
	0.0025	14	14	10	14	14	2	10	8	10			
	0.005	6	2	4	8	12	2	2	2	4			
	0.0075	47	6	8	4	14	8	6	10	8			
	0.01	57	8	2	8	8	12	10	14	33			
VoxelGrid	0	24	6	6	8	6	2	8	2	4			
	0.001	10	6	10	16	14	2	8	6	2			
	0.0025	10	10	14	8	8	2	4	2	4			
	0.005	2	8	8	6	14	8	4	8	4			
	0.0075	31	4	4	4	4	4	8	8	10			
	0.01	49	2	4	6	6	0	4	18	20			



**Fig. 8** Charts representing the mean results for experiment sets using the FPFH descriptor.

**Table 13** Results for the experiments using Spin Image and Harris 3D.

Models	Scene's Noise	Scenes									
		RAW	GNG				VoxelGrid				
		All Points	10000	15000	17500	20000	10000	15000	17500	20000	
RAW	0	12	10	4	2	8	8	4	8	0	
	0.001	12	4	2	6	12	12	0	8	0	
	0.0025	6	6	18	8	4	14	6	6	10	
	0.005	6	22	10	29	16	2	6	16	14	
	0.0075	18	12	12	18	31	6	2	4	4	
	0.01	51	12	14	10	18	6	6	12	14	
GNG	0	10	10	16	12	10	10	8	6	6	
	0.001	6	10	6	10	2	4	8	6	6	
	0.0025	2	4	6	10	6	10	10	14	8	
	0.005	2	10	10	14	18	6	2	10	14	
	0.0075	16	2	14	12	20	8	2	4	18	
	0.01	71	6	4	12	12	14	18	29	27	
VoxelGrid	0	6	24	12	2	8	4	6	6	4	
	0.001	4	10	8	10	6	4	4	8	6	
	0.0025	2	10	6	4	4	8	6	6	4	
	0.005	2	10	18	24	10	6	4	8	12	
	0.0075	20	12	12	18	24	10	6	6	18	
	0.01	57	2	6	6	10	4	8	6	27	

## 5.4 Discussion

These results support our proposal that the use of GNG improves the results of recognition in noisy point clouds. Comparing the results obtained with the descriptors, the highest recognition percentage for the GNG clouds was

**Table 14** Results for the experiments using Spin Image and Intrinsic Shape Signature.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	63	2	0	2	6	2	0	2	0
	0.001	71	2	0	4	2	2	8	8	6
	0.0025	39	0	0	6	2	2	0	4	6
	0.005	6	0	2	6	2	0	6	4	2
	0.0075	10	0	0	4	6	2	2	2	0
	0.01	47	2	2	6	0	2	2	0	2
GNG	0	37	0	2	8	2	0	0	0	4
	0.001	53	0	6	2	2	2	2	0	4
	0.0025	29	0	4	8	0	0	2	4	8
	0.005	4	0	4	2	4	0	0	0	4
	0.0075	51	0	2	2	2	0	0	2	0
	0.01	82	0	0	0	0	2	2	0	0
VoxelGrid	0	27	2	4	4	2	12	4	2	4
	0.001	63	0	2	6	4	0	6	0	6
	0.0025	35	0	0	6	6	4	2	4	16
	0.005	18	0	2	4	2	0	2	4	2
	0.0075	55	0	0	2	0	0	0	2	0
	0.01	82	0	0	0	0	0	2	0	0

**Table 15** Mean values of the results for the experiments using Spin Image.

		Scenes								
		RAW	GNG				VoxelGrid			
Models	Scene's Noise	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	0	33	7	6	3	7	6	2	4	2
	0.001	33	5	3	7	7	5	3	10	3
	0.0025	17	3	11	10	5	6	2	5	7
	0.005	4	10	5	14	10	2	5	8	9
	0.0075	14	7	5	11	17	5	4	3	1
	0.01	46	6	6	9	11	7	6	7	10
GNG	0	22	5	10	10	9	5	3	2	5
	0.001	23	5	8	8	5	2	4	5	4
	0.0025	15	6	7	11	7	4	7	9	9
	0.005	4	4	6	8	12	3	1	4	7
	0.0075	38	3	8	6	12	5	3	5	9
	0.01	70	5	2	7	7	10	10	14	20
VoxelGrid	0	19	11	7	5	5	6	6	3	4
	0.001	26	5	7	11	8	2	6	5	5
	0.0025	16	7	7	6	6	5	4	4	8
	0.005	7	6	10	12	9	5	3	7	6
	0.0075	35	5	5	8	10	5	5	5	10
	0.01	63	1	3	4	5	1	5	8	16

**Table 16** Mean values of the results for the experiments using the evaluated keypoint detectors with the Spin Image descriptor, grouped by the number of representatives.

	Scenes								
	RAW	GNG				VoxelGrid			
	All Points	10000	15000	17500	20000	10000	15000	17500	20000
RAW	25	6	6	9	10	5	4	6	5
GNG	29	5	7	8	9	5	5	7	9
VoxelGrid	28	6	7	8	7	4	5	5	8

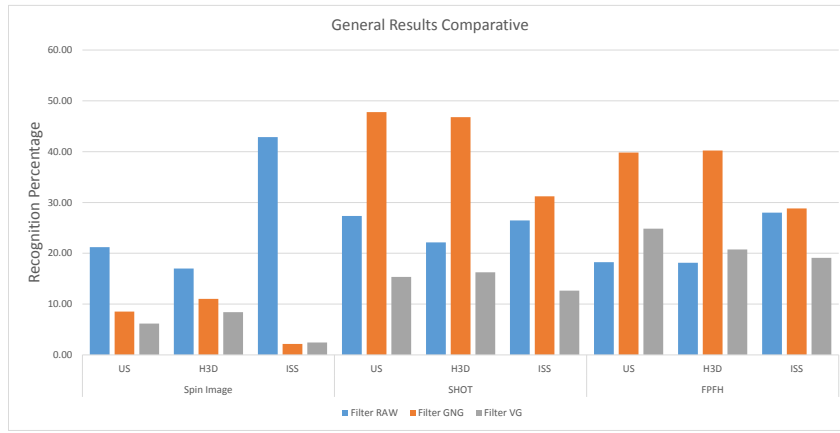
achieved by the SHOT feature descriptor with 41,9%, followed by FPFH with 36,2% and, finally, Spin Image with 7,2%.

Table 17 and Figure 9 show a comparison of the different detectors, descriptors and filter methods. The best recognition rate is achieved by the combination of Uniform Sampling and SHOT, using GNG with 17500 neurons.

The Spin Image descriptor cannot be used with a noise reduction method due to its way of building the descriptor. Furthermore, this descriptor is computationally expensive.

**Table 17** Results for the experiments by detector, descriptor and filter.

Feature Descriptor	Keypoint Detector	Filter		
		RAW	GNG	VG
Spin Image	US	21.20	8.53	6.18
	H3D	17.01	11.03	8.42
	ISS	42.86	2.15	2.44
SHOT	US	27.35	47.79	15.35
	H3D	22.14	46.78	16.25
	ISS	26.47	31.22	12.64
FPFH	US	18.25	39.82	24.83
	H3D	18.14	40.22	20.75
	ISS	28.00	28.83	19.10



**Fig. 9** Results for the experiments by detector, descriptor and filter.

In SHOT and FPFH with GNG and in Spin Image with the raw data, the recognition rate is lower for clouds without noise than with noise. This is due to the way the descriptor is calculated. For SHOT and FPFH with the raw data, the behavior is as the expected: the more noise there is, the lower the recognition rate becomes. But GNG is able to capture the topology of the 3D data and this topology is well adapted to both descriptors. However, for Spin Image and the raw data, the descriptor does not adapt to the topology provided by the GNG and, thus, the recognition rate is almost negligible.

## 6 Conclusions

The presence of noise in a 3D point cloud reduces the performance in a recognition process. Therefore, it is necessary to use a method to reduce the noise and, therefore, to increase the recognition rate. Our results show that the use of the GNG method improves recognition rates, obtaining better results than those for Voxel Grid and for the raw clouds. GNG reduces the noise without losing significant information, and enables good recognition results. In addition, we identify that Uniform Sampling is the keypoint detector that achieves the best rates of recognition together with the SHOT feature descriptor.

As future work, we propose to continue evaluating more 3D detection and description methods to find better combinations of performance and noise robustness. We also plan to include RGB information in the GNG in order to give support to keypoints and descriptors which use color information.

## References

1. Aldoma, A., Tombari, F., Di Stefano, L., Vincze, M.: A global hypotheses verification method for 3d object recognition. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) *Computer Vision? ECCV 2012, Lecture Notes in Computer Science*, vol. 7574, pp. 511–524. Springer Berlin Heidelberg (2012)
2. Asari, M., Sheikh, U., Supriyanto, E.: 3d shape descriptor for object recognition based on kinect-like depth image. *Image and Vision Computing* **32**(4), 260 – 269 (2014)
3. Besl, P., McKay, N.: A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (1992)
4. Chen, H., Bhanu, B.: 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters* **28**(10), 1252 – 1262 (2007)
5. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: G. Medioni (ed.) *1991 Proceedings., IEEE International Conference on Robotics and Automation*, 1991., pp. 2724–2729 vol.3 (1991)
6. Computer Vision LAB: SHOT: Unique signatures of histograms for local surface description - computer vision LAB. URL <http://www.vision.deis.unibo.it/research/80-shot>
7. Fritzke, B.: A growing neural gas network learns topologies. In: *Advances in Neural Information Processing Systems 7*, pp. 625–632. MIT Press (1995)
8. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3d object recognition in cluttered scenes with local surface features: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **36**(11), 2270–2287 (2014)
9. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 858–865 (2011)
10. Johnson, A., Hebert, M.: Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing* **16**(9-10), 635 – 651 (1998)
11. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **21**(5), 433–449 (1999). DOI 10.1109/34.765655
12. Kohonen, T.: *Self-Organising Maps*. Springer-Verlag (1995)
13. Marius Muja: FLANN - fast library for approximate nearest neighbors : FLANN - FLANN browse. URL <http://www.cs.ubc.ca/research/flann/>
14. Martinetz, T.: Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. In: S. Gielen, B. Kappen (eds.) *Proc. ICANN'93, Int. Conf. on Artificial Neural Networks*, pp. 427–434. Springer, London, UK (1993)

15. Martinetz, T., Schulten, K.: Topology representing networks. *Neural Networks* **7**(3) (1994)
16. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **36** (2014)
17. Pang, G., Neumann, U.: Training-based object recognition in cluttered 3d point clouds. In: *3D Vision - 3DV 2013, 2013 International Conference on*, pp. 87–94 (2013). DOI 10.1109/3DV.2013.20
18. PCL: Documentation - point cloud library (PCL). URL [http://pointclouds.org/documentation/tutorials/normal\\_estimation.php](http://pointclouds.org/documentation/tutorials/normal_estimation.php)
19. Radu Bogdan Rusu: Point cloud library (PCL): pcl::UniformSampling< PointInT > class template reference. URL [http://docs.pointclouds.org/1.7.0/classpcl\\_1\\_1\\_uniform\\_sampling.html#details](http://docs.pointclouds.org/1.7.0/classpcl_1_1_uniform_sampling.html#details)
20. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 3212–3217 (2009)
21. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer* **27**(11), 963–976 (2011). DOI 10.1007/s00371-011-0610-y
22. Tombari, F., Di Stefano, L.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In: *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pp. 349–355 (2010)
23. Tombari, F., Gori, F., Di Stefano, L.: Evaluation of stereo algorithms for 3d object recognition. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 990–997 (2011)
24. Tombari, F., Salti, S.: A combined texture-shape descriptor for enhanced 3d feature matching. In: *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 809–812 (2011)
25. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III, ECCV'10*, pp. 356–369. Springer-Verlag, Berlin, Heidelberg (2010)
26. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision* **102**(1-3), 198–220 (2013). DOI 10.1007/s11263-012-0545-4. URL <http://dx.doi.org/10.1007/s11263-012-0545-4>
27. Viejo, D., Garcia, J., Cazorla, M., Gil, D., Johnsson, M.: Using GNG to improve 3d feature extraction-application to 6dof egomotion. *Neural Networks* (2012)
28. Xu, G., Mourrain, B., Duvigneau, R., Galligo, A.: Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design* **45**(2), 395 – 404 (2013). *Solid and Physical Modeling 2012*
29. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 689–696 (2009)